Bordeaux INP
ENSEIRB-MATMECA

Luc Mieussens
24 septembre 2019

PETIT GUIDE POUR EMACS

Ce guide est destiné aux débutants sous Emacs. Il présente le fonctionnement de base d'Emacs et donne quelques commandes usuelles. Des références plus détaillées sont données à la fin du guide.

Table des matières

1 Introduction

1.1 Avantages d'Emacs

Malgré un aspect assez rebutant au premier abord, vous découvrirez vite qu'Emacs est un éditeur de textes très agréable et très puissant.

Tout d'abord, un grand nombre de commandes usuelles (déplacement, copier/coller, recherche/remplacement) peuvent être exécutées rapidement en utilisant des raccourcis clavier, qui s'apprennent assez facilement. Quelques conseils pratiques pour faciliter la frappes de certaines séquences seront données dans des rubriques intitulées **TRUCS** à la fin des sections.

Ensuite, un des grands avantages d'Emacs est la possibilité pour l'utilisateur de fabriquer ses propres commandes de traitement de texte (voir la section 5.1), ce qui est extrêmement utile pour écrire des programmes ou pour le traitement de textes *via* LaTeX par exemple.

1.2 Conventions utilisées dans ce guide

Elles sont celles d'Emacs pour la représentation des séquences de touches à frapper, écrites par convention avec la police `Teletype` :

- `RET` : frappe de la touche `Enter` (ou `Entrée`).
- `DEL` : frappe de la touche `Delete` (ou `BackSpace` ou ←).
- `ESC` : frappe de la touche `Esc`.
- `ESPACE` : frappe de la touche `ESPACE`
- `C-x` : frappe de la touche `Ctrl` et de la touche `x` simultanément.
- `M-x` : frappe différée de la touche `Esc` puis de la touche `x`.
- `M-x toto` : frappe différée de la touche `Esc` puis de la touche `x` puis de la séquence de touches `toto` .

Dans tout ce document, les noms des commandes Emacs sont en *italique*. Les séquences de touches à frapper sont en `Teletype` (raccourcis clavier et commandes). Les commandes se lancent soit en frappant `M-x nom-de-la-fonction`, soit avec les raccourcis-clavier associés.

TRUCS : une séquence de touches du type `C-x` (avec un tiret) est obtenue en les frappant simultanément. En pratique, on laisse la touche `C` enfoncée pendant qu'on frappe `x`. Au contraire, une séquence du type `C r` (sans tiret) est obtenue en frappant les touches successivement.

1.3 Définitions

- Région : suite de caractères commençant et finissant à des positions données.
- Buffer : copie de travail d'un fichier (cette copie ne remplace le fichier que lors de la sauvegarde). Emacs peut en ouvrir plusieurs à la fois.
- Fenêtre : Zone de l'écran sur laquelle est affiché un buffer. On peut ouvrir plusieurs.
- Le presse-papiers (aussi appelé *kill-ring*) est une zone tampon où est expédié tout texte copié.

2 Organisation de l'écran

L'écran d'Emacs est découpé au démarrage en quatre zones distinctes :

- la barre de menu (*menubar*),
- la fenêtre (*window*),
- la ligne de mode (*mode line*),
- la zone d'écho (*echo area / minibuffer*)

La barre de menu permet d'accéder à des commandes utiles *via* la souris.

La fenêtre contient le buffer en cours d'édition. Voir la section 4.2 pour plus d'informations sur les fenêtres.

La ligne de mode contient des informations sur le buffer édité dans la fenêtre à laquelle elle appartient : le nom du buffer, les modes en cours, si le buffer contient des modifications non sauvegardées, et la position de la fenêtre dans le buffer.

La ligne d'écho se trouve en bas de la fenêtre. Elle est utilisée pour afficher le texte que vous saisissez lors d'une commande (*minibuffer*). Elle affiche de plus les messages d'erreur ainsi que tout le texte qu'Emacs renvoie lors d'une commande. Tous les messages affichés dans la ligne d'écho sont copiés dans un buffer (voir sec. 4.2) nommé `*Messages*`.

3 Commandes usuelles

3.1 Lancement et interruption d'Emacs

On lance Emacs en tapant la commande `emacs` dans le shell. Mais pour éditer le fichier *toto.txt*, on utilise la ligne de commande `emacs toto.txt`.

Pour quitter Emacs, taper la séquence `C-x C-c`.

3.2 Déplacement du curseur

Dans un premier temps, le plus simple est de déplacer le curseur à l'aide des flèches du clavier, ou de cliquer à la souris sur la nouvelle position désirée. Cependant, pour certains déplacements, il est souvent beaucoup plus rapide d'utiliser des raccourcis clavier. Par exemple, pour aller à la fin de la ligne, il suffit de taper `C-e`.

Voici les commandes de déplacement les plus utiles :

— `C-a` *(beginning-of-line)* : déplacement au début de la ligne courante.
— `C-e` *(end-of-line)* : déplacement à la fin de la ligne courante.
— `M-<` *(beginning-of-buffer)* : déplacement au début du buffer.
— `M->` *(end-of-buffer)* :déplacement à la fin du buffer.

Les deux dernières commandes seront avantageusement associées aux touches début et fin du clavier (cf. section 5.2).

Il existe aussi des commandes de déplacement d'un seul caractère, mais elles ne sont plus utiles aujourd'hui.

3.3 Copier/couper-coller

3.3.1 Effacement simple

Pour effacer du texte, le plus simple pour commencer est d'utiliser la touche `DEL`, ou `BackSpace` ou [←]. Cette touche détruit le caractère immédiatement à gauche du curseur.

Deux autres commandes simples sont aussi utiles :

— `C-d` *(delete-char)* : Détruit le caractère situé sous le curseur.
— `C-k` *(kill-line)* : Détruit tous les caractères du curseur jusqu'à la fin de la ligne.

3.3.2 Sélection, copie et effacement de régions

Pour sélectionner une région, il suffit de positionner le curseur au début ou à la fin d'une zone de texte puis de taper `C-ESPACE`. Emacs vous informe dans la zone d'écho que le marquage du début

de la zone est effectué (`mark-set`). Ensuite, positionner le curseur sur l'autre extrémité de la zone. Selon la configuration du fichier .emacs, la zone sélectionnée peut être visualisée en grisé.

Ainsi tous les caractères compris entre ces deux positions sont sélectionnés. Taper `C-x C-x` permet alors de ramener le curseur an début de région, celle-ci restant sélectionnée.

Ensuite, en tapant `C-w` ou `M-w`, le texte de la région est envoyé dans le presse-papiers (mais dans le cas de `C-w` le texte est effacé (texte 'coupé') et dans le cas de `M-w`, le texte reste dans le buffer édité (texte 'copié').

3.3.3 Le presse-papiers

Le texte du presse-papiers peut être rappelé en utilisant `C-y` *(yank)* , qui insère à la position du curseur le texte se trouvant dans le presse-papiers (c.-à-d. le dernier texte envoyé dans le presse-papiers).

Ainsi un copier-coller d'une région s'obtient après sélection en tapant `M-w` puis `C-y` à l'endroit souhaité, alors qu'un couper-coller s'obtient en tapant `C-w` puis `C-y`. Remarquer qu'on peut coller plusieurs fois le même texte du presse-papier.

3.4 Utilisation de la souris

Ce qui est décrit aux paragraphes précédents peut parfois être avantageusement exécuté avec la souris. Pour sélectionner une région : cliquer le bouton gauche de la souris au début de la région, puis cliquer le bouton droit de la souris à la fin de la région. Pour copier cette région, cliquer le bouton du milieu à l'endroit du texte souhaité (on aura effectué un copier-coller). Pour la détruire, cliquer une deuxième fois le bouton droit à la fin de la région après la sélection, on peut alors la coller comme précédemment (on aura effectué un copier-coller).

3.5 Copier/couper-coller des rectangles

Il peut être utile de pouvoir manipuler des régions rectangulaires, c.-à-d. des colonnes de texte (comme la colonne d'un tableau de chiffres), ce que les commandes précédentes ne permettent pas (car toute ligne sélectionnée est alors traitée jusqu'au dernier caractère de la ligne).

Il faut d'abord sélectionner le rectangle comme une zone normale au clavier ou à la souris : sélectionner le coin en haut à gauche de votre rectangle, puis le coin en bas à droite. Attention, la zone qui semble sélectionnée est plus grande que le rectangle car toutes les lignes comprises entre les deux points sont grisées. Cela n'est pas important : les commandes suivantes ne prendront en compte que le rectangle défini par les deux points sélectionnés :

— `C-x r k` (*kill-rectangle*) : effacement du rectangle et envoi dans le presse-papier

— `C-x r y` (*yank-rectangle*) : copie du rectangle du presse-papier à l'endroit souhaité

Pour copier sans détruire, annuler la première commande *kill-rectangle* (cf. section suivante), le rectangle est ainsi restitué, et il reste aussi dans le presse-papier pour la copie.

3.6 Annulation

La dernière action effectuée peut être annulée en tapant `C-x u` ou `C-_` *(undo)* .

En outre, on peut remonter dans la pile de commandes précédentes jusqu'à la première de la session ! On peut ainsi annuler les n dernières commandes effectuées. Pour cela, il suffit de taper `C-x u` ou `C-_` *(undo)* autant de fois que nécessaire.

Pour annuler un commande en cours (vous avez tapé une partie du nom de la commande, commencé à taper le nom d'un fichier, etc.), taper `C-g` *(keyboard-quit)* .

3.7 Manipulation de fichiers

Commandes de manipulation de fichiers :

- Ouverture : `C-x C-f` *nom-du-fichier*
- Création : Pareil mais *nom-du-fichier* est un fichier inexistant.
- Insertion : `C-x i` à l'endroit où l'on désire insérer l'autre fichier.
- Sauvegarde : `C-x C-s`
- Renommer : `C-x C-w` *nom-du-fichier*

Pour chacune de ces commande, Emacs demande le nom du fichier dans la zone d'écho.

TRUCS : de telles séquences (du type `C-x C-f`) sont faciles à taper en laissant la touche `Ctrl` enfoncée pendant que l'on tape successivement les touches `x` puis `f`.

3.8 Chercher et remplacer

3.8.1 Recherche incrémentale

La recherche incrémentale ("incremental search" en anglais) d'une chaîne de caractères se fait au fur et à mesure que l'utilisateur tapes la chaîne en question. Le curseur se place sur la première occurrence correspondant à la partie de la chaîne déjà tapée.

Pour lancer cette recerche, taper `C-s` *(isearch-forward)* puis votre chaîne de caractères (qui apparaît dans la zone d'écho). La recherche s'effectue vers l'avant. Pour une recherche vers l'arrière, taper `C-r` *(isearch-backward)*.

Arrivée en fin de texte, la recherche reprend au début.

Pour relancer la recherche précédente, taper deux fois `C-s`.

3.8.2 Chercher/Remplacer

Pour remplacer une occurrence dans un texte (comme le nom du variable par un autre dans un programme), il y a deux possibilités :

- pour remplacer toutes les occurrences : `M-x replace-string` *(replace-string)* suivie de la chaîne à remplacer. Valider, puis donner la chaîne de remplacement. Attention, ça peut être dangereux ...
- pour remplacer après confirmation : `M-%` *(query-replace)* suivie de la chaîne à remplacer. Valider, puis donner la chaîne de remplacement. Confirmer le remplacement de l'occurrence en tapant `ESPACE`, refuser en tapant `DEL`.

4 Buffers et Fenêtres

Quand on travaille sur un fichier avec Emacs, on travaille en fait sur une copie de ce fichier, appelée buffer. Le fichier n'est réellement modifié qu'à la sauvegarde du buffer. On peut ouvrir plusieurs buffers dans une seule session Emacs. Enfin, on peut découper l'écran d'Emacs en plusieurs fenêtres, ce qui permet de visualiser plusieurs buffers à la fois.

4.1 Utiliser les buffers

Quand on lance la commande `emacs toto`, Emacs s'ouvre directement en ouvrant un buffer copie du fichier *toto*. Quand on a plusieurs buffers ouverts, on peut en afficher la liste avec `C-x C-b` (cette liste est en fait elle-même un buffer nommé `*Buffer List*`). On peut aussi voir cette liste en tapant `CTRL` bouton gauche de la souris.

Pour fermer le buffer courant, taper `C-x k` *(kill-buffer)* .

Pour sauver le buffer courant taper `C-x C-s`.

4.2 Utiliser les fenêtres

Au démarrage, Emacs ouvre une fenêtre pour le buffer du fichier donné à la commande `emacs`.

Les commandes pour manipuler les fenêtres sont les suivantes :

— `C-x 2` *(split-window-horizontally)* divise horizontalement la fenêtre courante en deux fenêtres contenant chacune le buffer courant.

— `C-x 3` *(split-window-vertically)* divise verticalement la fenêtre courante en deux fenêtres contenant chacune le buffer courant.

— `C-x o` *(other-window)* le curseur passe dans la fenêtre suivante.

— `C-x 0` *(delete-window)* détruit la fenêtre courante.

— `C-x 1` *(delete-other-windows)* détruit les autres fenêtres.

Ces commandes sont très importantes. Elles permettent de visualiser plusieurs fichiers à la fois avec une seule session Emacs.

5 Personnaliser Emacs

Une fois qu'on sait utiliser facilement Emacs, il peut être intéressant de le personnaliser en fonction de ses propres besoins (programmation, traitement de textes etc.). Ceci est possible en créant ses propres commandes (les macros), en associant des commandes usuelles aux touches du clavier, ou en modifiant des variables d'Emacs. Les modifications peuvent être restreintes à la session courante d'Emacs, ou être permanentes (il faut alors modifier le fichier de paramètres .emacs qui se trouve sous votre répertoire home).

5.1 Les Macros

Une macro est une suite de commandes Emacs enregistrée par l'utilisateur, et exécutable par la suite comme une commande Emacs existante.

Ces macros permettent de faire très rapidement de grosses modifications sur un texte en quelques secondes. C'est extrêmement utile pour écrire des programmes ou pour le traitement de textes *via* LaTeX par exemple.

Voici tout de suite un exemple : on veut une commande qui parcourt tout le texte, et qui pour toute chaîne contenue entre deux accolades, la remplace par la chaîne ***, va à la ligne et écrit %, et enfin sauve le fichier. On commence la définition de la macro avec `C-x (` , puis on tape les commandes nécessaires (elles agissent directement sur le texte pendant la définition de la macro) : recherche d'une accolade ouvrante `C-s {` , puis marquage du début de la zone `C-ESPACE`, puis recherche de l'accolade fermante suivante `C-s }` , puis destruction de la région sélectionnée entre les accolades `C-w`, puis on tape ***, on va à la ligne avec `RET`, on tape %, on sauve le fichier `C-x C-s` et enfin on termine la définition de la macro avec `C-x )`. Ensuite, on peut l'utiliser avec `C-x e` ou les autres commandes ci-dessus.

Voici les commandes pour utiliser les macros :

— `C-x (` : *(start-kbd-macro)* Commence la définition d'une macro.

— `C-x )` : *(end-kbd-macro)* Finit la définition d'une macro.

— `C-x e` : *(call-last-kbd-macro)* Exécute la dernière macro définie.

— *name-last-kbd-macro* : nomme la dernière macro définie (essentiel pour la sauver).

— *insert-kbd-macro* : insère une macro dans un fichier.

— `M-x` *nom de la macro* : exécute la macro *nom de la macro* .

Ainsi, une macro nommée *couac* avec *name-last-kbd-macro* et placée dans le .emacs avec *insert-kbd-macro* sera utilisable dans n'importe quelle session Emacs, comme n'importe quelle commande (avec `M-x` *couac*).

Attention, si une erreur intervient durant la définition de la macro, celle-ci sera inutilisable et il faut la redéfinir.

5.2 Personnalisation du clavier

On peut associer des commandes (ou des macros) à des touches du clavier, ou à associer des commandes à des séquences de touches (c.-à-d. créer des raccourcis clavier). Pour cela, il faut écrire des instructions en langage LISP dans le fichier `.emacs`. Voici deux instructions utiles qui vous éviteront d'apprendre le LISP :

— pour associer une touche à une commande ou à un séquence de touches :
(define-key global-map touche 'commande)
ou *(define-key global-map touche "séquence-de-touches")*

— pour associer une séquence de touches *séquence1* à une commande ou à un séquence de touches *séquence2* :
(global-set-key "séquence1" 'commande)
ou *(global-set-key "séquence1" "séquence2")*

Dans ces instructions, les touches spéciales sont notées ainsi :

— `\C-x` : touche simultanée de `CTRL` et de la lettre `x` (raccourci noté `C-x` dans ce guide).
— `\e` : touche `ESC`.
— `\r` : touche `RET`.
— `\t` : touche `TAB`.
— `[f1] [f2] ... [f12]` : touches fonctions F1 F2 ... F12

Exemples :

1. pour écrire la chaîne `_pr` en tapant la touche F5, insérer dans le `.emacs` la ligne :
   `(define-key global-map [f5] "_pr")`
2. pour associer la commande `goto-line` (pour aller à une ligne donnée) à la séquence `C-x g`, insérer dans le `.emacs` la ligne :
   `(global-set-key "\C-xg" 'goto-line)`
   (ce raccourci clavier est très utile et souvent présent, mais pas toujours, dans le `.emacs` initial.)

5.3 Les variables d'Emacs

Toutes les variables d'Emacs peuvent être modifiées (pour la session en cours ou de façon permanente (par exemple la variable `fill-column` qui définit la colonne à partir de laquelle la ligne est coupée). Voir le manuel GNU d'Emacs pour la liste complète de ces variables.

Pour modifier une variable, taper `M-x customize nom-variable`, puis suivre les instructions.

6 Les modes d'Emacs

Emacs possède différents modes associés aux différents types de fichiers que l'on utilise. Par exemple, les modes texte, latex, ou f90 sont respectivement associés aux fichiers de texte quelconques, aux fichiers LaTeX, et aux fichiers Fortran 90.

Ces modes rendent actifs un certains nombre de commandes, raccourcis clavier, mise en page automatique, etc., uniquement pour les fichiers auquel le mode est destiné.

Par exemple, quand on ouvre un fichier Fortran 90, le mode f90 est activé, ce qui permet d'utiliser les possibilités suivantes :

— taper `C-l` met en valeurs des mots clefs du Fortran 90 (couleur et police)
— taper `TAB` sur une ligne indente automatiquement cette ligne, ou la complète (*end* complété par *end if, end do, end program* etc. selon les cas).
— taper `C-;` pour commenter une région sélectionnée

Le mode est activé en fonction de l'extension du fichier (.txt, .tex, .f90, etc.). Pour activer un autre mode nommé toto, utiliser la commande *toto-mode*.

7 Note sur la barre de menu d'Emacs

Nombre de commandes citées précédemment sont utilisables en cliquant à la souris dans le menu, comme dans la plupart des éditeurs de texte. Cependant, il est plutôt conseillé d'apprendre les commandes usuelles et leurs raccourcis-clavier (chargement et sauvegarde de fichier, copier/couper-coller, déplacement début-fin de ligne, etc.), car le gain de temps lors de la frappe du texte est alors considérable.

8 Aide d'Emacs

Dans le menu help de la barre de menu, on trouvera une liste des raccourcis, la description d'un raccourci ou d'une fonction particuliers, une liste de paramètres modifiables, etc.

9 Références

Je me suis beaucoup inspiré de deux manuels de Benjamin Drieu qui étaient disponibles sur `http://echo-linux.alienor.fr/articles/doc-emacs/emacs.html` et `http://www.april.org/groupes/doc/emacs/emacs.html` mais qui semblent introuvables à l'heure actuelle.

Le guide de référence est évidemment celui de GNU, disponible en ligne et en anglais sur `http://www.gnu.org/software/emacs/manual`. Ne l'imprimez pas, il est extrêmement volumineux.

Vous trouverez le début de la traduction du manuel sur
`http://www.linux-france.org/article/appli/emacs/manuel/`

Un autre site utile en français :
`https://www.emacswiki.org/emacs/PlanDuSite`